Docket No.: GE04347 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 000043471
James T. Lynn et al.	:	Confirmation Number: 3710
Application No.: 09/814,601	:	Tech Center Art Unit: 2137
Filed: March 23, 2001	:	Examiner: Zachary A. Davis

For: METHOD FOR SECURELY DISTRIBUTING SOFTWARE COMPONENTS ON A COMPUTER NETWORK

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

The real party in interest is Motorola, Inc.

(2) Related Appeals and Interferences

None.

(3) Status of Claims

Rejected Claims

1-5.

Claims Appealed

1-5.

(4) Status of Amendments

The amendment filed on March 9, 2005, in response to the non-final Office Action dated September 9, 2004 was entered by the final Office Action dated April 19, 2005. The amendment filed on September 19, 2005, in response to the Advisory Action dated August 5, 2005 was entered by the non-final Office Action dated December 21, 2005.

(5) Summary of Claimed Subject Matter

1. A method for securely distributing a component from a network host to a network appliance, comprising the steps of:

signing, by said network host, a configuration file including a load table which defines a plurality of authorized components for said network appliance (*See e.g.*, Application at page 2, lines 7-9 and 18-20);

executing a secure kernel and said signed configuration file on said network appliance, said secure kernel including computer code for checking the authenticity of said configuration file and boot code for allowing said network appliance to initially boot up and establish communication with said network host (*See e.g.*, Application at page 2, lines 9-13; page 3, lines 2-10; and FIG. 1);

verifying, by said secure kernel, the authenticity of said configuration file *(See e.g.*, Application at page 2, lines 25-26; page 3, lines 4-8; and FIG. 1);

reading, by said secure kernel, said load table only after said verifying step (*See e.g.*, Application at page 3, lines 16-20 and FIG. 1); and

loading said plurality of authorized components defined in said load table onto said network appliance (*See e.g.*, Application at page 3, lines 16-24 and FIG. 1).

(6) Grounds of Rejection to be Reviewed on Appeal

The rejection of claims 1-5 as being as anticipated under 35 U.S.C. § 102(e) by U.S. Patent Number 6,049,671 ("Slivka").

(7) Argument

The following remarks address the improper rejections of the claims under 35 U.S.C. § 102(e). The Final Office Action has failed to establish a *prima facie* case of anticipation. MPEP § 2131 states that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either

expressly or inherently described, in a single prior art reference." Appellant respectfully submits that Slivka fails to describe or suggest each and every element of independent claim 1. In addition to failing to describe each and every element of independent claim 1, Slivka also fails to describe or suggest each and every element of dependent claim 5. The subsequent paragraphs first address the improper rejection of independent claim 1, and then address the improper rejection of dependent claim 5.

A. Slivka fails to describe or suggest all the features of independent claim 1 and therefore fails to anticipate this claim and its dependent claims

Claims 1-5 were rejected under 35 U.S.C. § 102(e) as being anticipated by Slivka. Appellant respectfully requests reversal of this rejection for at least the following reasons.

Claim 1 recites a method for securely distributing a component from a network host to a network appliance. The method include steps of signing, by said network host, a configuration file including a load table which defines a plurality of authorized components for said network appliance and executing a secure kernel and said signed configuration file on said network appliance. The secure kernel includes computer code for checking the authenticity of said configuration file and boot code for allowing said network appliance to initially boot up and establish communication with said network host. The method also includes steps of verifying, by said secure kernel, the authenticity of said configuration file; reading, by said secure kernel, said load table only after said verifying step; and loading said plurality of authorized components defined in said load table onto said network appliance.

To provide context, in one aspect, the application provides a method for securely distributing software components in a network environment. Application at page 2, lines 6-7. Toward this end, a secure kernel and a configuration file containing a load table are initially loaded onto each network appliance. Application at page 2, lines 7-9. The secure kernel includes the minimum amount of boot code for allowing the network appliance to initially boot up and establish communication with the network host. Application at page 2, lines 9-11. The secure kernel also contain a security mechanism, such as an algorithm or other device for verifying the authenticity of the configuration file associated with the network appliance. Application at page 2, lines 11-13.

In a preferred embodiment, the configuration file associated with each network appliance is digitally signed or otherwise encoded by the network host to ensure the authenticity of the load table within the configuration file. Application at page 2, lines 18-20. If the authenticity of the

configuration file is confirmed, the secure kernel reads the load table from the configuration file and loads and initiates the appropriate software components (e.g., a paid television program) as defined by the load table. Application at page 3, lines 16-20. Alternatively, if the authenticity check of the configuration file fails, the secure kernel logs this failure and sends a request to the host for a new configuration file. Application at page 3, lines 10-12. In this manner, the user who has tampered with the configuration file in an attempt to obtain unauthorized access to a program, an application, or the like will be recognized and his/her attempt will fail. Application at page 3, lines 12-15.

Appellant respectfully requests reversal of the above-stated rejection because Slivka, at a minimum, fails to describe or suggest a method for securely distributing a component from a network host to a network appliance, the method including, among other steps, a step of signing, by said network host, a configuration file including a load table which ***defines a plurality of authorized components for said network appliance***, as recited in claim 1 (emphasis added).

Slivka, in FIG. 2, discloses a system for providing computer software from an update service computer 40 to a user computer 34. Slivka at Abstract and col. 5, line 66 to col. 6, line 4. Referring to FIG. 3 of Slivka, running on update service computer 40 is a service update application 48 that will communicate with a user update application 50 on user computer 34. *Id*. To access update service computer 40, a user starts user update application 50 on user computer 34, which establishes a two-way communication link 36 with update service computer 40. Slivka at col. 6, lines 12-15. Thereafter, update service computer 40 receives from user update application 50 an inventory of computer software stored on user computer 34. Slivka at col. 5, lines 47-52. The update service computer 40 compares the received inventory with database entries on update service computer 40. *Id*.

After the comparison, server update application 48 sends back a summary of available computer software to user update application 50, which displays them to the user. Slivka at col. 5, lines 55-57. The summary includes information such as, for example, availability of patches and fixes for existing computer software, brand new computer software, and new versions of existing computer software. Slivka at col. 5, lines 57-60 and col. 8, lines 33-38. The user may then select to download one of the available items listed on the summary. Upon selecting a particular item (e.g., an enhanced version of a network browser), service update application may create a cabinet file to transfer a selected item to a user by providing appropriate packaging for a secure transfer, and adding an installation program to make the download program self extracting when received by user computer 34. *See e.g.*, Slivka at col. 13, lines 6-68; col. 17, line 51 to col. 18, line 48; and FIG. 7.

To this end, Slivka appears to be nothing more than an on-line store to allow a user to pick and choose any program desired regardless of whether or not the network appliance is authorized to use the program. That is, Slivka is not concerned whether or not the network appliance itself is authorized to operate a program. As such, Slivka does not describe or suggest a method for securely distributing a component from a network host to a network appliance, the method including, among other steps, a step of signing, by said network host, a configuration file including a load table which ***defines a plurality of authorized components for said network appliance***, as recited in claim 1 (emphasis added).

The Office Action asserts that the cabinet file in Slivka corresponds to the configuration file recited in claim 1. *See e.g.*, Final Office Action dated May 26, 2006 at page 5, lines 6-7. Appellant disagrees because, at a minimum, the alleged cabinet file does not include a load table which defines a plurality of authorized components for said network appliance (emphasis added). The Office Action appears to recognize this shortcoming and thus points to an entirely different section of Slivka (e.g., column 8, lines 34-43) to show that the alleged cabinet file (appearing for the first time on column 13, line 45) includes a load table, allegedly defining a plurality of authorized components for said network appliance. *See e.g.*, Final Office Action dated May 26, 2006 at page 5, lines 7-8.

In column 8, lines 34-43, however, Slivka merely shows a list of available computer software that can be updated on the user's computer should the user desire the same. To this end, the user is asked to choose which available computer software shown on the summary should be downloaded. Slivka at col. 8, lines 42-45. Upon selecting a particular item, the service update application may create the alleged cabinet file to transfer a selected item to a user by providing appropriate packaging for a secure transfer, and adding an installation program to make the download program self extracting when received by user computer. *See e.g.*, Slivka at col. 13, lines 6-68; col. 17, line 51 to col. 18, line 48; and FIG. 7. For example, if the user selects to download an enhanced version of network browser, the service update application creates the alleged cabinet file for the enhanced version of the selected software. *Id*.

As such, the cabinet file is created after the alleged summary of available computer software for download is presented to the user. Therefore, the cabinet file cannot include the alleged summary of available computer software. Furthermore, there is no suggestion to include such summary (e.g., menu of available programs, new programs and updates discussed in column 8 of Slivka) in the alleged cabinet file, as the Office Action concludes. As a practical matter, it would hardly be desirable or

beneficial to encrypt a listing of programs to prevent viewing by the public at large when one is trying to sell such programs to the public at large. Moreover, while Slivka clearly discloses to install the program in the cabinet file on a user's computer, a listing of programs and updates available for downloading, which is likely to frequently change, clearly would not be beneficial or desirable to have installed on a user's computer.

For at least these reasons, Slivka fails to describe or suggest a method for securely distributing a component from a network host to a network appliance, the method including, among other steps, a step of signing, by said network host, a configuration file including a load table which ***defines a plurality of authorized components for said network appliance***, as recited in claim 1 (emphasis added).

Accordingly, Appellant respectfully requests reversal of the rejections of claim 1, along with its dependent claims.

B. Slivka fails to describe or suggest all the features of dependent claim 5.

Appellant respectfully submit that the dependent claim 5 is also allowable on its own merit. Claim 5 recites a method for securely distributing a component from a network to a network appliance, the method includes, among other steps, a step of generating, by said host, an updated configuration file and signing, by said host, said updated configuration file (emphasis added). Neither of these features are disclosed by in Slivka. In particular, Slivka does not describe updating the configuration file and signing the updated configuration file, including a load table which defines the programs a network appliance is authorized to use.

The Office Action takes an inconsistent position in the rejection claim 5. In particular, in rejecting claim 5, the Office Action asserts the summary list (not the cabinet file) described in column 8, lines 27-33 corresponds to the updated configuration file. *See e.g.*, Final Office Action dated May 26, 2006 at page 5, lines 22-24. Appellant disagrees at least because the alleged summary list is not signed.

The Office Action asserts that Slivka discloses this feature in column 16, lines 65-67. Appellant again disagrees. In column 16, lines 65-67, Slivka describes a SIGNMS application configured to append a digital signal to the self-extracting executable distribution file. The self-extracting executable distribution file is composed of the cabinet file and a self-extracting application

called "WEXTRACT.EXE," and it is not composed of the alleged summary list. As such, in this portion Slivka describes signing the alleged cabinet file and not the alleged summary list.

For at least the foregoing reasons, Slivka fails to describe or suggest a method for securely distributing a component from a network to a network appliance, the method including, among other steps, a step of generating, by said host, an updated configuration file and signing, by said host, said updated configuration file, as recited in claim 5 (emphasis added).

Applicants, Appellant respectfully request reversal of the rejections of claim 5.

The brief fee of $500 is authorized to be charged to Deposit Account No. 06-1050. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: June 16, 2008

/Larry T. Cullen/
Larry T. Cullen
Registration No. 44, 489

Motorola Connected Home Solutions
101 Tournament Drive
Horsham, PA 19044
Phone: 215-323-1797

Appendix of Claims

1. (Previously presented) A method for securely distributing a component from a network host to a network appliance, comprising the steps of:

signing, by said network host, a configuration file including a load table which defines a plurality of authorized components for said network appliance;

executing a secure kernel and said signed configuration file on said network appliance, said secure kernel including computer code for checking the authenticity of said configuration file and boot code for allowing said network appliance to initially boot up and establish communication with said network host;

verifying, by said secure kernel, the authenticity of said configuration file;

reading, by said secure kernel, said load table only after said verifying step; and

loading said plurality of authorized components defined in said load table onto said network appliance.

2. (Original) The method of claim 1, wherein said loading step comprises loading an operating system.

3. (Original) The method of claim 1, wherein said loading step comprises loading a computer software application.

4. (Original) The method of claim 1, wherein said loading step comprises loading services.

5. (Original) The method of claim 1, further comprising the steps of:

generating, by said host, an updated configuration file;

signing, by said host, said updated configuration file;

transmitting said signed updated configuration file from said host to said network appliance;

verifying, by said secure kernel, the authenticity of said updated configuration file; and

thereafter reading, by said secure kernel, said updated configuration file.

Evidence Appendix

None.

Related Proceedings Appendix

None.